SpeakerID - Voice Activity Detection

Victor Lenoir

Technical Report n^o1112, June 2011
revision 2288

Voice Activity Detection has many applications. It's for example a mandatory front-end process in speech encoding and speech/speaker recognition. This report presents two different algorithms for voice activity detection (VAD): one using thresholds and the second one using gaussian mixture models (GMM).
The proposed algorithms use short-term features such as short-term Energy, spectral flatness measure or Mel-Frequency Cepstral Coefficient (MFCC). The different VAD algorithms are compared in different noise environments in order to highlight their noise robustness.

La détection de voix a de nombreuses applications. C'est par exemple une étape obligatoire avant de faire de la reconnaissance du locuteur. Ce rapport présente deux différents algorithme pour la détection de voix (VAD) : un utilisant des seuils et le second utilisant des mélanges de gaussiennes (GMM).
Les algorithmes proposés utilisent des caractéristiques calculées sur des petits intervalles de temps comme par exemple l'énergie, la monotonie spectral ou les Mel-Frequency Cepstral Coefficients (MFCC). Les différents algorithmes de détection de voix sont comparés dans différentes conditions de bruit afin de mettre en évidence leur robustesse aux bruits.

Keywords
voice activity detection, gaussian mixture model, cepstral vectors, speaker recognition

Laboratoire de Recherche et Développement de l'Epita
14-16, rue Voltaire – F-94276 Le Kremlin-Bicêtre cedex – France
Tél. +33 1 53 14 59 47 – Fax. +33 1 53 14 59 22
lenoir@lrde.epita.fr – http://www.lrde.epita.fr/

Copying this document

Contents

Chapter 1

Introduction

Voice activity detection (VAD) is used to filter the audio file in order to select only the segments where there is speech.
Nowadays, voice activity detection has many applications: in speech encoding, audio conferencing and in particular in speech and speaker recognition. With a voice activity detection the data used in the speaker recognition system are more relevant than raw data with silents and noises.
In this paper, I present two voice activity detection algorithms. One using gaussian mixture model proposed by Patric Kenny and Senoussaoui (2010), and the other one using thresholds.

I present these algorithms because they have a very similar structure and used the same initialization procedure. Hence one of the only difference is the model used to store the speech and noise model.

Chapter 2 of this report describes the features used, the noise and the applications of the voice activity detection.
Chapter 3-4 of this report describes these two algorithms and discuss about their advantages and disadvantages.
Finally in chapter 5, We connect the voice activity detection algorithms to the rest of the speaker recognition process to estimate the quality of each voice activity detection algorithms plotting the DET curves.

Chapter 2

Prerequisites

2.1 Voice Activity Detection

2.1.1 Description

Voice Activity Detection is a technique used to detect human speech in an audio recording. The idea is to separate speech segment from silence and noise. It has a wide application in voice communication. (used in GSM for example)

One of the principal assumption made used in VAD algorithms is that the speech is the energetically dominant signal. This is generally true for male speech signal. Indeed the energy is lower in female speech signal.

VAD has many applications, it's currently used in hands-free telephony in order to delete noise. But also in:

- Speaker recognition
- Speech encoding
- Audio conferencing

2.1.2 Noise

Noise can be defined as the contamination of the desired signal by another unwanted signal.

The purpose of the Voice activity detection is to get rid off the noise and the silent (low-energy) segments. There are different types of noise, associated with a colour. For example white noise is a noise with a flat spectrum.

2.2 Feature Extraction

2.2.1 Cepstral Feature

In order to detect speech in an audio recording, we have to extract some features to analyse them. In most of speech recognition's algorithms, we promote cepstral coefficients unlike raw signal because the cepstral coefficients have more meanings in speech recognition.

The cepstral coefficients are obtained by doing a Fourier Transformation (we use a short-term analysis). Therefore we have to define a window in order to process the signal in discret time and thus keep a time data to possibly do the segmentation. (Figure 2.2)

These parameters are based on a short-term analysis using a sliding analysis window. A feature vector is extracted for each placement of that window. The Mel Frequency Cepstral Coefficients (MFCCs Figure 2.1) are the most important parameters used to represent speaker vocal track characteristics, they are heuristic representations of Acoustics propertiesm that simulate the human ear.
In our case the cepstral coefficients are extracted with the HTKtool HCopy. We use a window of 25ms with a 10ms shift, we extract the energy and 19 cepstral coefficients, their derivative and their acceleration.
And we finally have a sequence of 60-dim Cepstral vectors.

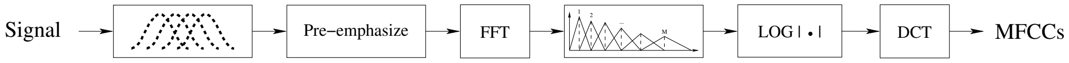

Figure 2.1: Mel-Frequency Cepstral Coefficients extraction process

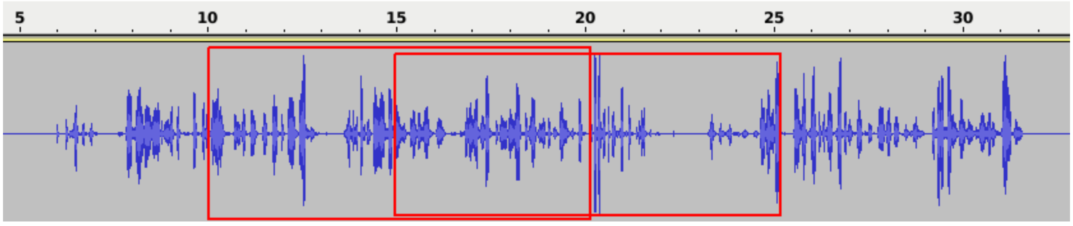

Figure 2.2: Cepstral coefficients extraction with a window size of 10ms and a 5ms shift

2.2.2 Statistical Feature

In addition to cepstral coefficients and energy we used other features which highlight the difference between speech and noise

Feature	Domain	Description
Zero Crossing Rate	Time	Rate of sign-changes along a signal
Spectral Flatness Measure	Frequency	Coefficient indicating the flatness of the spectrum
Signal Flatness Measure	Time	Coefficient indicating the flatness of the signal

The relevance of each features depend on SNR (Signal to Noise ratio).
Figures 2.3, 2.4 and 2.5 show the plotting of a few features computed on an extremely white noised signal. (signal is in blue and feature in red).
We can see that some features are not affected as much as other by noise.

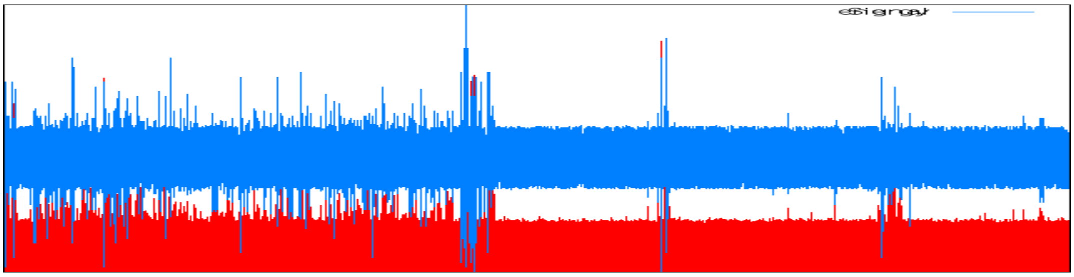

Figure 2.3: Energy plotting

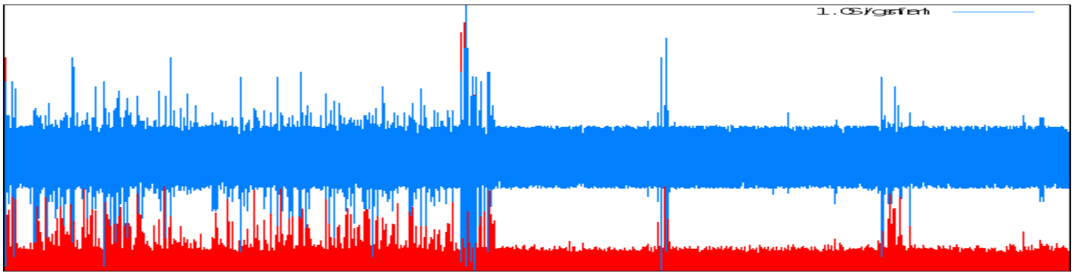

Figure 2.4: Inverse of spectral flatness measure plotting

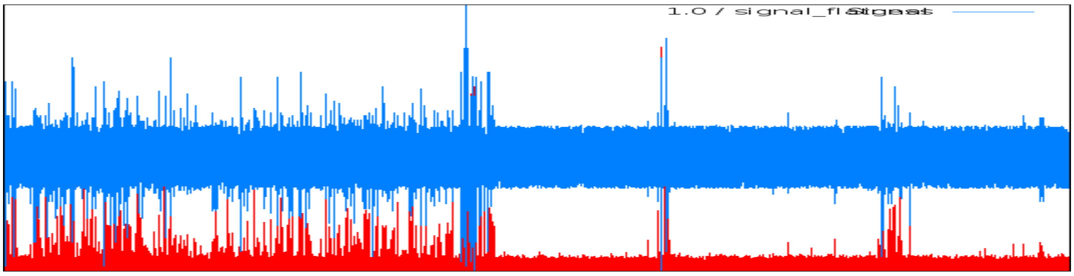

Figure 2.5: Inverse of signal flatness plotting

Chapter 3

Voice Activity Detection using Thresholds

Thresholds based VAD algorithm is the more simple and less complex method. Many version of this algorithm are used in practice.
We present in this chapter a version with an adaptative threshold in order to be more adapted with low SNR (Signal to Noise Ratio) as well as high SNR signal.
This algorithm can be decomposed in 4 phases:

1. The features extraction and main feature computation
2. We initialize the 2 thresholds which respectivelty represent very simple speech and noise model.
3. We learn more relevant thresholds with the initial thresholds on the whole signal
4. We marked as speech the frames closer to speech threshold than to noise threshold

3.1 Algorithm

Features Extraction We first extract the wanted features from the signal.
For each frame X we compute a new feature Δ using the basic features:

$$\Delta(X) = \frac{F_I(X) * E(X) * F_A(X)}{\delta + SFM(X) + Z(X)}$$

Where:

- $F_I(X)$ Max Frequency
- $F_A(X)$ Max Frequency Amplitude
- $E(X)$ Energy
- $Z(X)$ Number of zero crossing
- $SFM(X)$ Spectral Flatness Measure
- $\delta = 10^{-6}$ in order to prevent division by 0

Initialization We use 10% of the frames of the whole signal having the highest Δ value in order to calculate the initial speech threshold. The Speech threshold is only the arithmetical mean of the Δ of these frames.
We do the same with 10% of the frames having the lowest Δ value in order to compute the noise threshold.

Learning Once we have the primitive thresholds. We can now learn better thresholds.
For each frame X, if their feature $\Delta(X)$ is nearest to the speech threshold than the noise threshold then the speech threshold is updated as below:

$$T_{n+1} = \frac{n * Tn + \Delta(X)}{n+1}$$

Where:

- T_{n+1} is the new threshold
- T_n is the actual threshold
- $\Delta(X)$ is the feature of frame X
- n is the number of frame used to computed the actual threshold

We do the same for the noise threshold when the feature $\Delta(X)$ is nearest to the noise threshold than the speech threshold. We repeat this procedure until convergence.

Clustering Once we have the final thresholds. We can now cluster our frames.
For each frame X, if their feature $\Delta(X)$ is nearest to the speech threshold than the noise threshold then we mark the frame as speech. We mark the frame as noise otherwise.

3.2 Output

Figure 3.1 and 3.2 show an example of segmentation obtained with this algorithm.

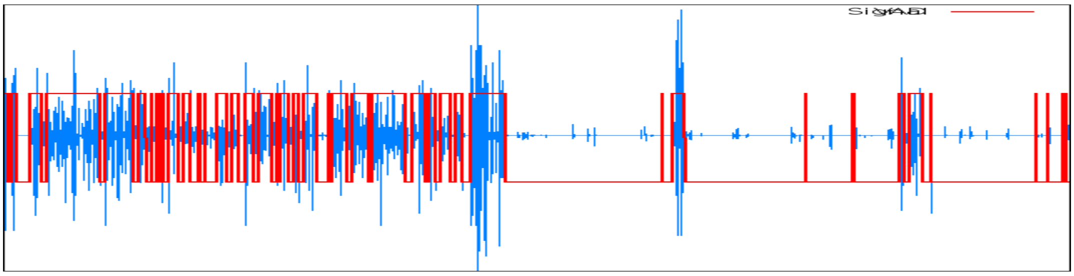

Figure 3.1: Segmentation obtained by threshold Voice Activity Detection

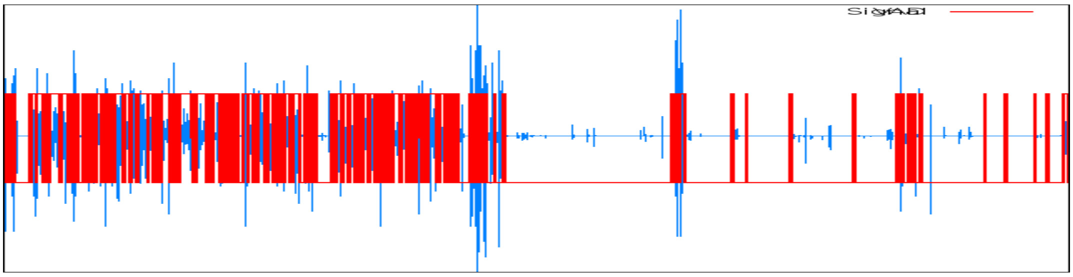

Figure 3.2: Reference Segmentation

3.3 Discuss

Advantages This algorithm is easy to implement and is less complex than the others
And it can obtained good result if the Δ formula is well chosen.

Disadvantages We make the asumption that there is at least 30% of speech in the signal. If there is no speech the procedure will fail.
Furthemore this algorithm can not be applied on-the-fly.
To solve theses problems a solution would be to fix the thresholds but the algorithm won't be as robust to SNR alteration.

Chapter 4

Voice Activity Detection using Gaussian Mixture Model

The structure of this algorithm (see Patric Kenny and Senoussaoui (2010)) is quite similar to the previous one. But instead of storing the speech and noise model in simple thresholds we use probabilistic model. This algorithm can be decomposed in 4 steps:

1. The features extraction and main feature computation
2. We initialize the 2 gaussian mixture models on a part of the signal
3. We learn new gaussian mixture models on the whole signal
4. We marked as speech the frames which belongs more likely to the speech model than the noise model

Figure 4.1 shows the algorithm flowchart.

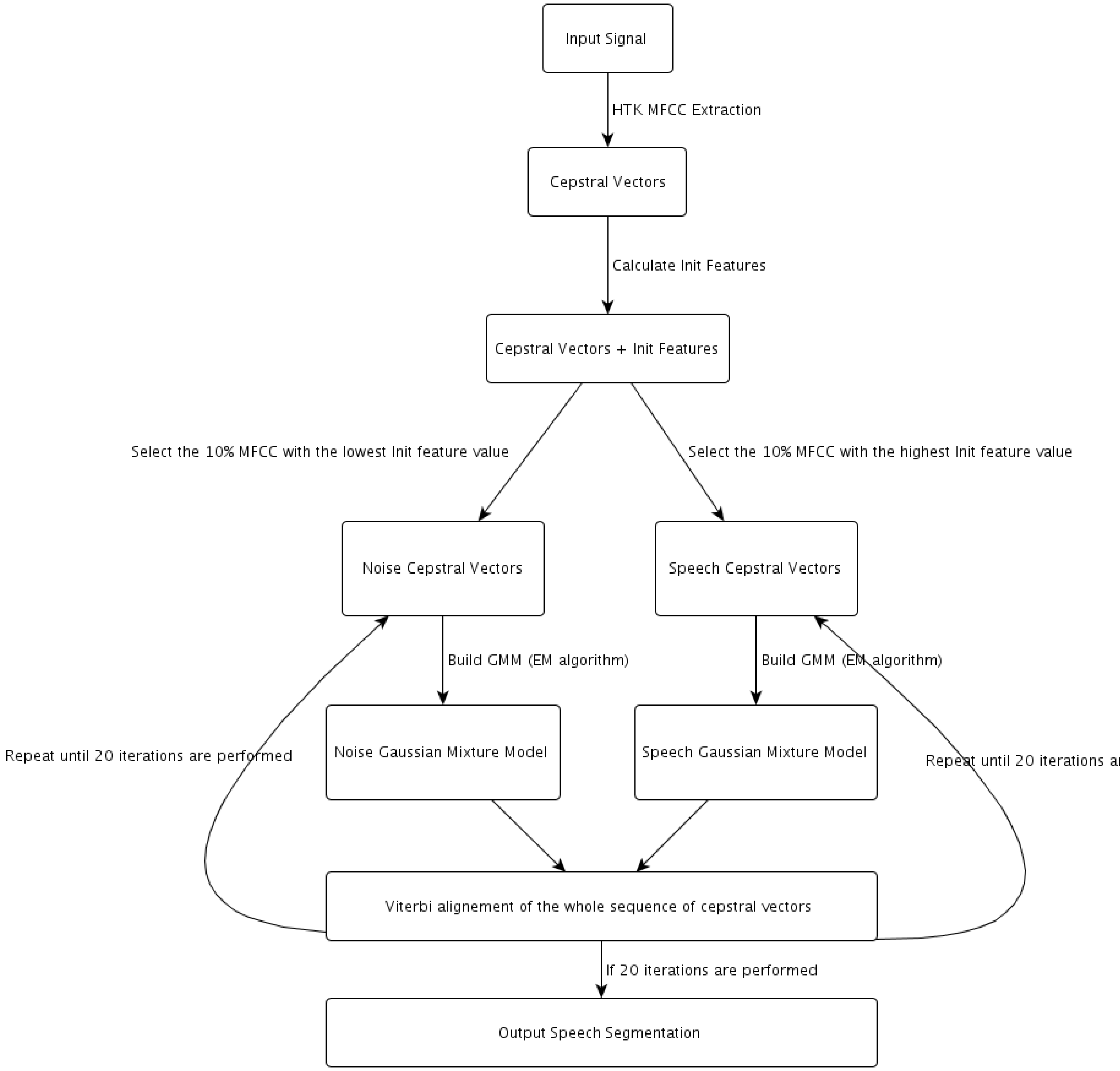
Input Signal
HTK MFCC Extraction
Cepstral Vectors
Calculate Init Features
Cepstral Vectors + Init Features
Select the 10% MFCC with the lowest Init feature value
Select the 10% MFCC with the highest Init feature value
Noise Cepstral Vectors
Speech Cepstral Vectors
Build GMM (EM algorithm)
Build GMM (EM algorithm)
Repeat until 20 iterations are performed
Noise Gaussian Mixture Model
Speech Gaussian Mixture Model
Repeat until 20 iterations a
Viterbi alignement of the whole sequence of cepstral vectors
If 20 iterations are performed
Output Speech Segmentation

Figure 4.1: Voice Activity Detection Algorithm Flowchart

4.1 Gaussian Mixture Model

Gaussian Mixture Model (GMM) is a probabilistic model used to represent a probability distribution defined as below:

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \Sigma_i)$$

Where

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} exp\{-\frac{1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i)\}$$

Where:

- $wi \geq 0$, $\sum_{i=1}^{M} w_i = 1$
- x is the data vector
- λ is the GMM set of parameters w_i, μ_i and Σ_i for $1 \leq i \leq M$
- M is the number of gaussians component
- w_i is the weight of the i^{th} gaussian component
- μ_i is the mean vector of the i^{th} gaussian component
- Σ_i is the covariance matrice of the i^{th} gaussian component
- D is the dimension of the x data vector

The GMM are trained using the Expectation Maximization algorithm (A.P. Dempster and Rubin (1977)).
In our algorithm, the x data vector is a 60-dim cepstral feature vector and we use this model to represent the probability distribution for noise and for speech.

4.2 Algorithm

Features Extraction We first extract the MFCCs (Mel-Frequency Cepstrum Coefficients).
In our case the cepstral coefficients are computed with the HTKtool HCopy. We use a window of 25ms with a 10ms shift and we extracte 19 cepstral coefficients plus the energy, their derivative and their acceleration.
We finally have a sequence of 60-dim Cepstral vectors.

Initialization For each fame X we calculate an additional feature in order to select trivial segment used for the initialization of our two gmms.
The initialization feature must highlight the differences between speech and noise.
In our case we choose the same formula as the first algorithm so that we can really compare theses algorithms:

$$\Delta(X) = \frac{F_I(X) * E(X) * F_A(X)}{\delta + SFM(X) + Z(X)}$$

Where:

- $\Delta(X)$ Initialization Feature of frame X
- $F_I(X)$ Max Frequency
- $F_A(X)$ Max Frequency Amplitude
- $E(X)$ Energy
- $Z(X)$ Number of zero crossing
- $SFM(X)$ Spectral Flatness Measure
- $\delta = 10^{-6}$ in order to prevent division by 0

We then take the 10% frames with highest initialization feature to build the speech gaussian mixture model.
Similarly we take the 10% frames with lowest initialization feature to build the noise gaussian mixture model.

Training To build a gaussian mixture model we first use the Linde-Buzo-Gray (Linde Y. and R.M. (1980)) algorithm to spread the 60-dim cepstral vectors in different gaussians.
Then we use the EM (Expectation-Maximization) algorithm to initialize each gaussian with theirs associated cepstral vectors.
In our case, we use 16 gaussian components for speech GMM and 4 gaussian components for noise GMM.

Clustering Once we have the two GMMs, we can now compare, for each frames, if the frame is more likely a speech-frame or a noise-frame by doing a Viterbi alignement.
With the resulting segmentation we can once more create two GMMs and repeat this procedure until convergence.

Output We now print the speech-frame segmentation of the last Virterbi alignement performed.
We also apply some operations in order to give consistence to the final segmentation. (like merging the close speech frame)

4.3 Output

Figure 4.2 and 4.3 show an example of segmentation obtained with this algorithm.

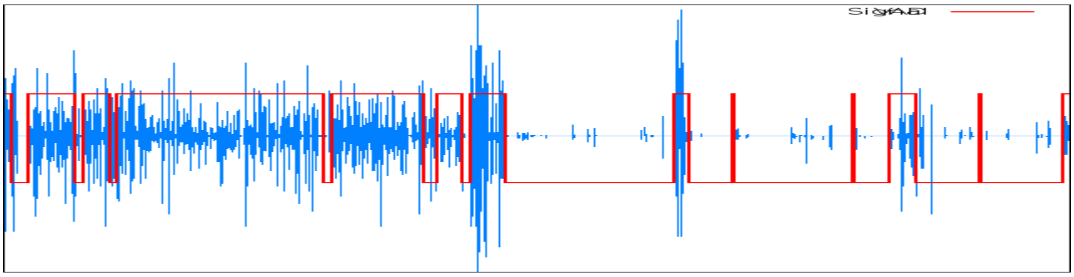

Figure 4.2: Segmentation obtained by Voice Activity Detection using Gaussian Mixture Model

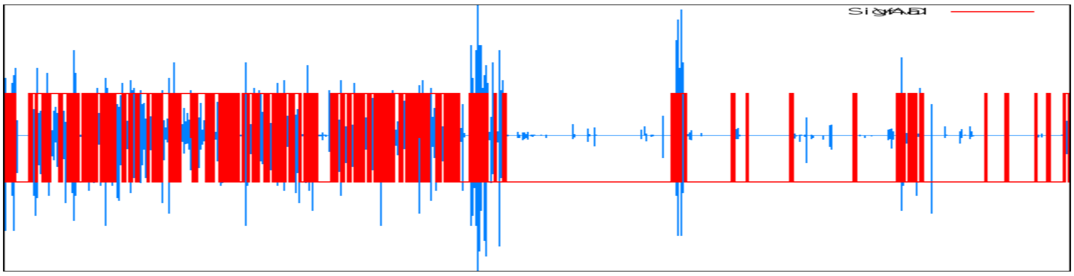

Figure 4.3: Reference Segmentation

4.4 Discuss

Advantages This algorithm can be easily modified to obtain better results, only by modifying the initialization feature formula.
Moreover it takes into account more than 60 features so it's more robust than an algorithm using only the energy.

Disadvantages This algorithm is complex and cannot be used on-the-fly.
Moreover if the audio file is full of noise/silence or full of speech. The procedure will fail, because we assume that there is at least 10% of noise and 10% of speech in the audio file.

Solution A solution to these disadvantages would be to train an Universal Speech GMM and an Universal Noise GMM using a great number of audio file.
In this case the algorithm will be usable on-the-fly and won't be as complex. Furthemore we won't have to make the assumption anymore that there is at least 10% of noise and 10% of speech in the signal.

Chapter 5

Benchmark

We use the LRDE speaker recognition system proposed to NIST-SRE 2010 to evaluate the relation between our VAD system and the Speaker recognition method. The speaker recognition method operates on cepstral features, extracted using a 25ms Hamming window. 19 mel frequency cepstral coefficients together with log energy are calculated every 10 ms. Delta and double delta coefficients were then calculated using a 5 frame window to produce 60-dimensional feature vectors. This 60-dimensional feature vector was subjected to feature warping using a 3s sliding window. We remove no speech frame before the processing. We used a gender dependent UBMs containing 2048 Gaussians (more detail about speaker recognition system can be found in

All our experiments were carried out on the extended trials of the core condition of the NIST 2010 SRE. We use only telephone and microphone conversation data of 5 minutes. A comparison of the results using our two vad systems with the original VAD system is reported in Table 5.1. The original VAD system uses an Hungarian speech recognizer to label speech frame This system was provided by Brno University.

To estimate the quality of a Voice Activity detection algorithm we connect the VAD algorithms to the rest of the speaker recognition process.
To evaluate performance of a speaker recognition, we plot the DET (Detection Error Tradeoff) curve (A. Martin and Przybocki (1999)) which is the miss probability in function of the false alarm probability.
An ideal DET curve intersects a point with very low false alarm probability and very low miss probability.

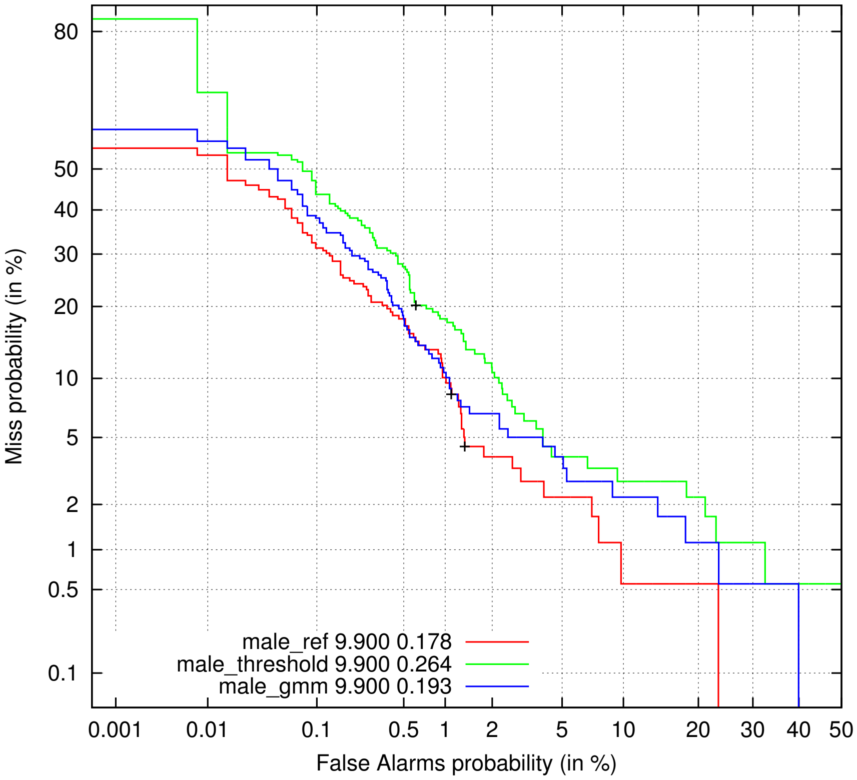

Figure 5.1: VAD algorithms Benchmark for Male audio files

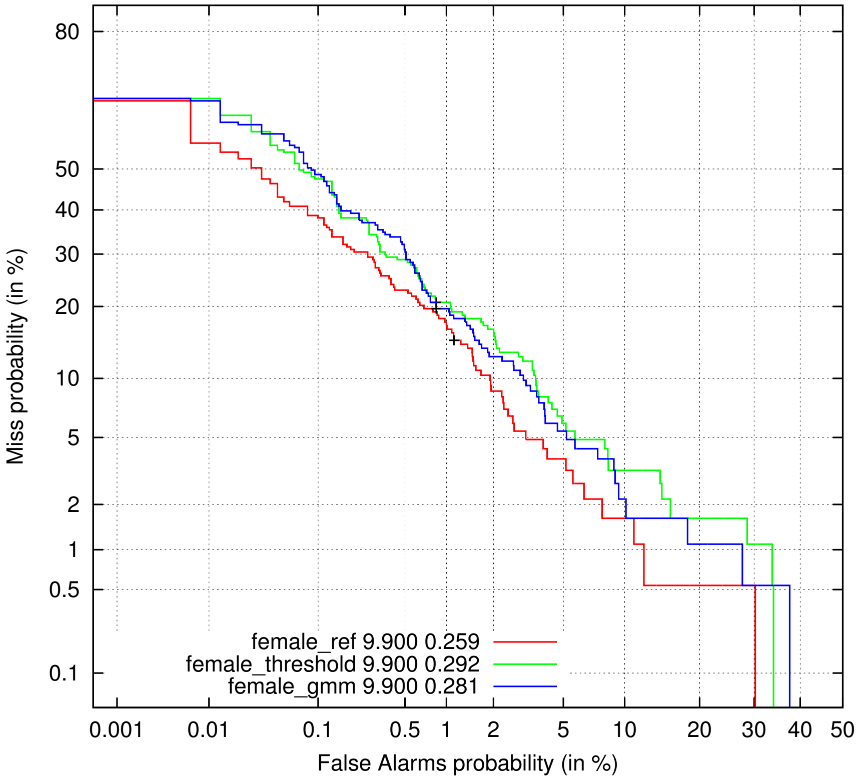

Figure 5.2: VAD algorithms Benchmark for Female audio files

We can observe in these benchmarks that we almost obtain the same results that the reference. Especially for the Gaussian Mixture Model VAD algorithm. But the reference still obtains better results than the two algorithms.
We also see that all of the VAD algorithms obtains worse results for female audio files than for male audio files. It can be explain by the similarities between female voice and noise.
Furthermore we can see that the GMM VAD algorithm obtain better results than the threshold VAD algorithm. It can be explain by two reasons:

- The MFCCs used in the GMM VAD are more characteristic than the computed feature
- The GMMs are more adapted than the threshold

Chapter 6

Conclusion

This report describes two different voice activity detection algorithms with their evaluation using them in a speaker recognition in order to plot DET curve.

Achieved work The benchmark shows that the voice activity detection algorithm using gaussian mixture models is almost as effective as the reference voice activity detection we already have. But the benchmarks were done with an old formula (only using the energy) and with a dilation too high. Indeed to increase the segmentation relevance, we did some post-treatment such as merging the closest frames and dilating the segments.
So we can expect better results with a good configuration.

Futur works I still have to evaluate the two algorithms with different configurations. Furthermore I have to try another type of algorithm which uses higher-order statistics (Michael Yaw Appiah and Munagala (2005)).

Chapter 7

Bibliography

A. Martin, G. Doddington, T. K. and Przybocki, M. (1999). The det curve in assessment of detection task performance.

A.P. Dempster, N. L. and Rubin, D. (1977). Maximum-likelihood from incomplete data via the em algorithm.

Douglas A. Reynolds, T. F. Q. and Dunn, R. B. (2000). Speaker verification using adapted gaussian mixture models.

Linde Y., Buzo, A. and R.M., G. (1980). An algorithm for vector quantizer design.

Michael Yaw Appiah, Raimonda Makrickaite, M. G. and Munagala, S. (2005). Robust Voice Activity Detection and Noise Reduction Mechanism using Higher-Order statistics.

Moattar, H. and Homayounpour, M. (2009). A simple but efficient real-time voice activity detection algorithm.

Patric Kenny, P. O. and Senoussaoui, M. (2010). The CRIM system for the 2010 NIST Speaker Recognition Evaluation.